

FIG. 1

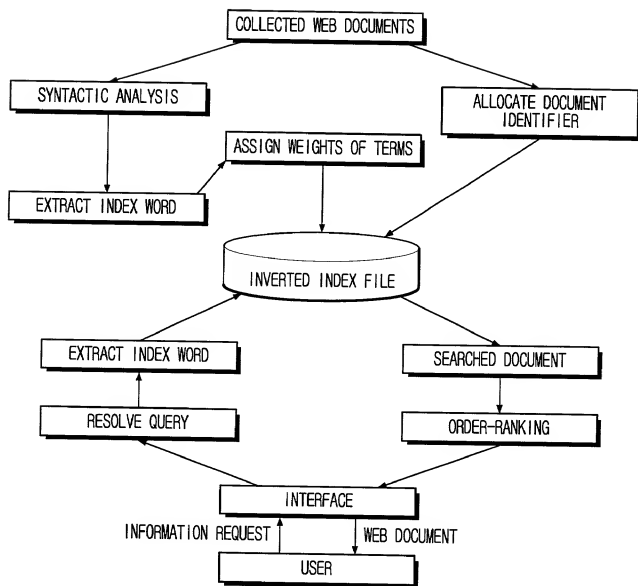


FIG. 2

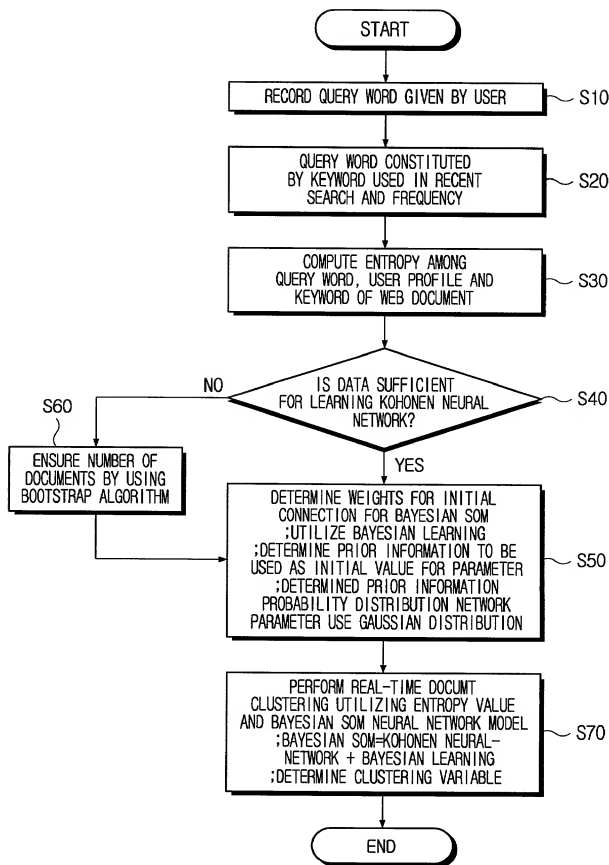


FIG. 3

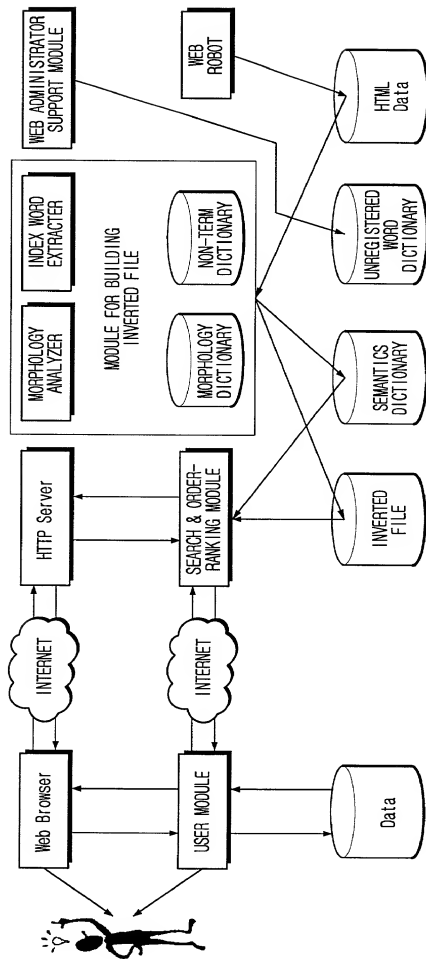


FIG. 4

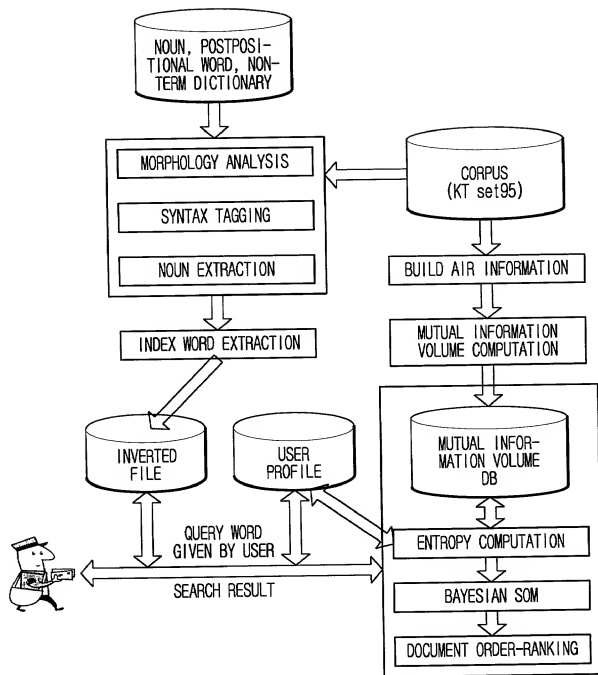


FIG. 5A

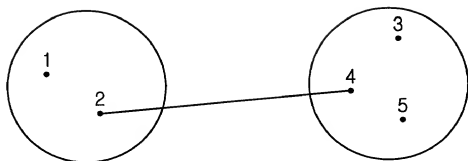


FIG. 5B

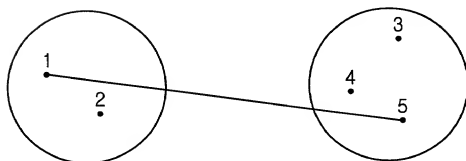


FIG. 5C

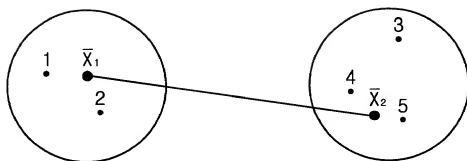


FIG. 5D

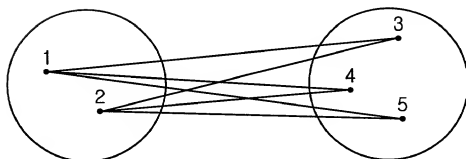


FIG. 6

```
Algorithm ClusteringofDocs(UserQryProfile[N], Ret_Docs[N])
// COMPUTE ENTROPY BY USING USER PROFILE AND KEYWORDS EXTRACTED
// FROM EACH DOCUMENT, AND PRODUCE DOCUMENT CLUSTER ACCORDING
// TO SIMILARITY
Set i, j, k to 0
for i = 1 to NumofRetDocs
  for j = 1 to NumofQuery
    for k = 1 to NumofTerms
      DocMatrix[i][j] = CalcEntropy(UserQryProfile[j], Ret_Docs[k]);
// COMPUTE P-NUMBER OF ENTROPY (KEYWORD, USER PROFILE), AND OBTAIN
// MATRIX HAVING SIZE OF N x P
Call CalcSim(Return SimDoc[NumofRetDocs], DocMatrix[j+k]);
// COMPUTE DISTANCE MATRIX HAVING SIZE OF N x N BETWEEN N-NUMBERS
// OF DOCUMENTS
for i = 1 to NumOfRetDocs
  Call CreatCluster(Return DocCluster[NumofCluster], SimDoc[i]);
// FORM CLUSTER BASED ON DISTANCE MATRIX
for j = 1 to NumOfCluster
  Call CalcSim(UserQryProfile[NumofCluster], DocCluster[j])
// OBTAIN DEGREE OF SIMILARITY BETWEEN EACH CLUSTER AND QUERY WORD
// GIVEN BY USER, AND EACH CLUSTER AND USER PROFILE
End ClusteringofDocs
```

FIG. 7

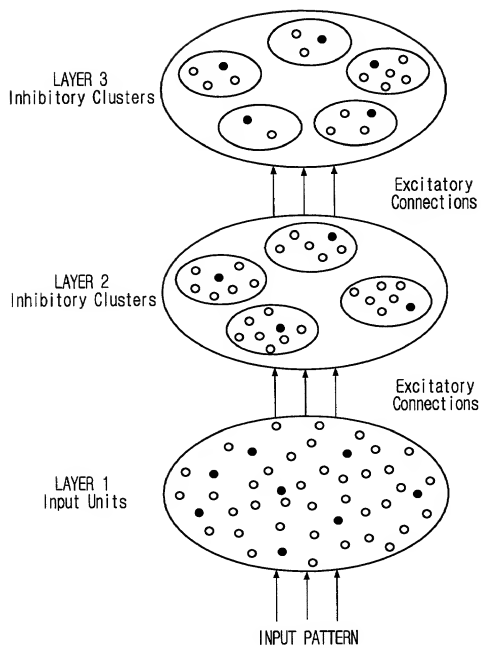


FIG. 8

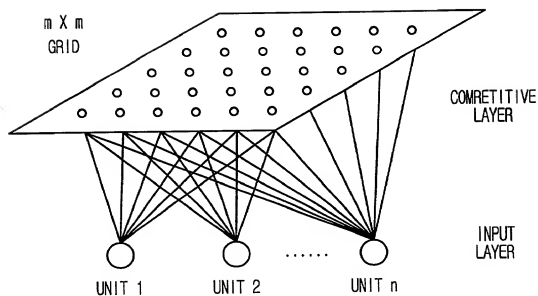


FIG. 9A

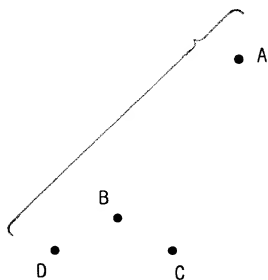


FIG. 9B

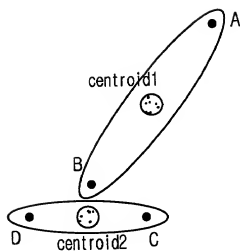


FIG. 9C

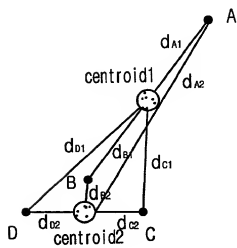


FIG. 9D

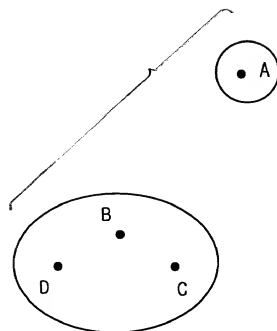


FIG. 10

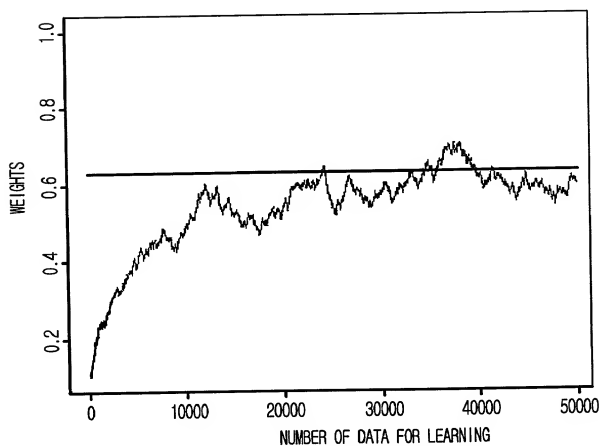


FIG. 11

```

Algorithm RankOfCluster(Ret_Docs[N])
// DOCUMENT CLUSTER BY BAYESIAN SOM AND ORDER-RANKING ALGORITHM
set i, j, k to 0;
for i = 1 to k;
    for j = 1 to 3;
        Index_Vector[k][j] = ExtrOfIndex(Ret_Docs[k]);
        Call MutualInformation(User_Q[NumofQuery], Index_Vector[k][j]);
        DocEntropyVector[i][j] = CalculateEntropy(Ret_Docs[k]);
    end j;
end i;
if NumofData <= 30 Call Bootstrap(DocEntropyVector[i][j]);
// PRODUCE SUFFICIENT DATA COLLECTION REQUIRED FOR LEARNING BAYESIAN
BAYESIAN NEURAL NETWORK BY EMPLOYING STATISTICAL BOOTSTRAP ALGORITHM
ALGORITHM IF DATA FOR LEARNING IS SMALL(FOR EXAMPLE, LESS THAN 30)
DecisionOfInitialWeight();
// DETERMINE INITIAL WEIGHTS FOR KOHONEN NETWORK BY UTILIZING PRIOR
DISTRIBUTION OF BAYESIAN. THAT IS, AVERAGE IS ZERO, AND INERSE NUMBER
OF SQUARE ROOT OF NUMBER OF NODES OF KOHONEN LAYER IS UTILIZED
AS STANDARD DEVIATION
Call BayesianSOM();
for i = 1 to NumOfCluster;
    CalculationOfNorm(Cluster[NumOfCluster]);
end i;
RankOfCluster(Value_of_Norm[NumOfCluster]);
// RE-RANK DOCUMENT CLUSTER HAVING HIGH SIMILARITY TO QUERY WORD
GIVEN BY USER
End Rank_of_Cluster
    
```